

Azure Dynamics Achieves 90% Code Reuse across Powertrain Control Projects with Model-Based Design

Azure Dynamics Corporation has established itself as a leading supplier of hybrid-electric and electric vehicle powertrains by competing effectively in an industry characterized by evolving customer requirements, rapidly changing technology, and regulatory guidelines for safety-critical systems. Their customer list includes London Taxis International, Canada Post, Purolator Courier Ltd., and other companies that are embracing electric/hybrid-electric solutions.

Azure achieves reusability to meet varying customer needs by using MathWorks tools for Model-Based Design to model, simulate, and automatically generate code for their powertrain control system applications.

“We place a high level of importance on the reusability and configurability of our software,” explains Michael Schaefer, senior engineer at Azure Dynamics. “With Model-Based Design, we reuse our models and fully automate production code generation and integration. We can now deliver new embedded systems in weeks and provide our customers with new builds in hours.”

THE CHALLENGE

Azure needed an agile development process that enabled them to support multiple projects and pursue new business. Conventional hand-coding methods were not feasible in an environment in which speed and quality were top priorities. Quality is a key concern because many of Azure’s systems are safety-critical, drive-by-wire applications that control everything between the pedals under a driver’s feet to torque at the wheels.

“Because the technology underlying electric and hybrid-electric vehicles is relatively new,



London Taxis International hybrid vehicle.

many of our customers struggle with defining and providing requirements,” Schaefer notes. “As a result, we must develop flexible software that can accommodate changes to business and customer requirements without affecting the overall design.”

Azure needed to automate and accelerate their development process for getting from requirements to verified implementation.

THE SOLUTION

Azure engineers used MathWorks tools for Model-Based Design to develop a flexible, configurable architecture for their electric and hybrid-electric powertrain systems.

Azure first implemented this reusable architecture for a commercial electric milk delivery vehicle. Engineers used Simulink® to develop control algorithms and a controller model as well as a plant model that included the electric vehicle’s major components, including battery packs, inverters, and motor. The team then combined the plant and controller models and ran non-real-time simulations in Simulink to establish a baseline for subsequent calibrations.

To facilitate reusability, Azure then partitioned the controller model, putting all interfaces to the operator, plant, and environment (OPE) in a single layer. They then replaced the OPE

THE CHALLENGE

To develop a reusable, flexible architecture for rapidly developing hybrid and electric powertrains

THE SOLUTION

Use MathWorks tools for Model-Based Design to model, simulate, and automatically generate production code based on a reusable design

THE RESULTS

- Achieved 90% reuse across projects
- Implemented 50 change requests monthly
- Automatically generated 100% of production code

layer simulation with blocks that communicate directly with the target hardware, an MPC555-based automotive controller.

The team used Stateflow® to model aspects of the major control subsystems, including states that track whether the subsystem is off, starting up, enabled, shutting down, or faulted.

Using Real-Time Workshop® Embedded Coder, engineers fully automated a build process for generating production code from the Simulink control model and integrating the code with low-level device drivers before download to the PCM. They then developed a testing strategy that included structural and functional testing.

“We are achieving coverage metrics reserved for highly regulated industries,” Schaefer explains. “Without tools like Reactis and its integration with our Simulink and Stateflow models, we could not achieve the level of test coverage we expect of any software that we provide our customers.”

Engineers also generated code from the plant model in hardware-in-the-loop system testing, which enabled engineers without access to a vehicle to perform high-level verification for changes as well as new features. This capability further reduced the software defect rate.

By using MathWorks tools to implement a hardware-independent, reusable architecture and a fully automated development and verification process, engineers easily retarget the same application to various customers’ controllers using different compiler and debug toolchains.

“Not only is our process faster, but our quality in the preproduction stage is much higher, saving us rework,” Schaefer explains. “Faster builds with fewer process iterations—it doesn’t get much better than that.”

“If we removed MathWorks tools from our development process, we could not compete effectively in the marketplace. With MathWorks tools, we model, simulate, and generate code for powertrain systems that will serve as the basis for all our future applications.”

Michael Schaefer, Azure Dynamics Corporation

THE RESULTS

- **Achieved 90% reuse across projects.** “Over four weeks, we will deliver software for three prototypes. For the electric vehicles, we are reusing approximately 90% of our code,” says Schaefer. “We also expect to increase our code reuse across pure EV, series hybrid, and parallel hybrid to 90%.”
- **Implemented 50 change requests monthly.** “Using MathWorks tools, we implemented 50 change requests to a control system in one month, rapidly delivering the project once we got the customer requirements,” Schaefer explains. “MathWorks tools enable us to iterate very rapidly and respond to fluctuations in the design cycle.”
- **Automatically generated 100% of production code.** “With our modest team size and large amount of projects, hand-coding is simply not feasible,” says Schaefer. “Our goal for the company is to automatically generate 100% of the code. With Real-Time Workshop Embedded Coder, we are meeting that objective.”

To learn more about Azure Dynamics, visit www.azuredynamics.com

www.mathworks.com

APPLICATION AREAS

- Automotive
- Control design
- Hardware-in-the-loop testing
- Model-Based Design
- Production code generation
- Simulation

PRODUCTS USED

- Simulink
- Stateflow
- Stateflow Coder
- Real-Time Workshop
- Real-Time Workshop Embedded Coder